## REMARKS

Claims 1-61 have been presented for examination. Claims 1-4, 6-8, 13-15 and 17-24 have been rejected under 35 U.S.C. §103(a) as being obvious over U.S. Patent No. 6,308,322 to Serocki *et al.* in view of U.S. Patent No. 5,765,037 to Morrison *et al.* Claims 5, 12 and 16 have been rejected as being obvious over the patent to Serocki *et al.* in view of the patent to Morrison *et al.* further in view of U.S. Patent No. 6,272,599 to Prasanna. Claims 9-11 have been rejected as being obvious over the patent to Serocki *et al.* and Morrison *et al.* further in view of U.S. Patent Publication No. 2002/0078268 to Lasserre. Claims 25-27, 29, 32-36, 38-40, 46-50, 52-54, 60 and 61 have been rejected under 35 U.S.C. §102(e) as being anticipated by the patent to Serocki *et al.* Claims 28, 37, 45 and 51 have been rejected as being obvious over the patent to Serocki *et al.* in view of the patent to Prasanna. Claims 30, 31, 41-44 and 55-58 have been rejected as being obvious over the patent to Serocki *et al.* in view of the patent publication to Lasserre. Claim 59 has been rejected as being obvious over the patent to Serocki *et al.* and the patent publication to Lasserre further in view of the patent to Prasanna. Additionally, the drawings have been objected to because of their informality, the specification has been objected to because of an incorrect reference numeral, and claim 45 has been rejected under 35 U.S.C. §112 as being indefinite.

The disclosed embodiments of the invention will now be discussed in comparison to the applied references. Of course, the discussion of the disclosed embodiments, and the discussion of the differences between the disclosed embodiments and the subject matter described in the applied references, do not define the scope or interpretation of any of the claims. Instead, such discussed differences merely help the Examiner appreciate important claim distinctions discussed thereafter.

The disclosed invention is directed to a system and method for improving the efficiency of caching instructions executed by a processor. Rather than relying on the conventional approach of caching instructions based on either how frequently or how recently an instruction was executed, the disclosed caching system and method determines the cacheability of an instruction at compilation based on other factors. Determining the cacheability of an instruction at compilation allows cacheability to be based on such factors as whether the

instruction is likely to cause "thrashing," whether the instruction is for an interrupt-service routine, whether the instruction requires real-time interaction, etc. Based on this cacheability determination during compilation, the instructions are marked with a bit corresponding to the determination. During execution of the instructions, the cacheability bit is detected and used by a computer system to either cache or not cache the instruction.

It is important to understand that the disclosed invention is not directed to other techniques for improving the efficiency at which a computer system executes instructions. For example, branch prediction techniques have been developed to improve the efficiency at which instructions can be prefetched in a pipelined processing system. In a pipelined processing system, instructions are fetched from either cache or system memory before the instructions are to be executed. However, the efficiency of prefetching instructions can be compromised when a branching instruction is encountered because it may not be possible to determine which branch will be taken before the branching instruction is to be fetched. Branch prediction techniques have been developed to allow prefetching of instructions from one of the branches based on such factors as which branch was most recently taken or which branch was most frequently taken.

The disclosed invention is also quite distinct from techniques for re-ordering the instructions in a program so that they can be processed most efficiently. In instruction re-ordering, a program is analyzed to determine the manner in which the instructions will be executed, and the relative order of the instructions can be changed to allow the program to be executed more efficiently. For example, if an instruction repetitively calls a subroutine that is located in a different area of the program, the execution efficiency may be improved by relocating the subroutine closer to the calling instruction.

In applicant's disclosed method and system, the efficiency at which a program can be executed is not improved by either prefetching instructions using branch prediction techniques or reordering instructions in the program. Instead, applicant's disclosed method and system more efficiently caches instructions based on factors determined during compilation and marks the instructions accordingly.

The primary reference cited in the Office Action is the patent to Serocki *et al.* According to the Office Action, the Serocki *et al.* patent teaches cache circuitry including a cache memory and a processor that caches instructions "based on at least in part on cacheability

determinations made during compilation of the computer program (*e.g.* Fig. 4; *target address hints* 520-Fig. 5B; *compiler generates... address hints, stores hinted addresses in a cache*, col. 8, lines 14-41...." [Office Action]. This portion of the specification quoted in the Office Action has absolutely nothing to do with determining the cacheability of instructions. Instead, this portion of the specification discusses the branch prediction techniques described in the Serocki *et al.* patent. Specifically, this portion of the specification states "once the rule-based most-likely target addresses are predicted, the compiler generates rule-based indirect *branch target address hints* and inserts them 520 into the intermediate code in advance of their corresponding indirect branches." [Col. 8, lines 14-18, emphasis added]. The Serocki *et al.* patent thus describes inserting branch target address hints into a program stored in cache memory 160 that correspond to the branches that the program is expected to take. The portion of the specification referring to cache memory 160 in line 37 simply states that the branch target address hints cannot be stored in cache memory too far in advance of the instructions for the branch because they might otherwise be overwritten with other instructions. The Examiner has not cited any portion of the Serocki *et al.* patent, and none is believed to exist, where the cacheability of instructions is discussed. Therefore, the Serocki *et al.* patent clearly does not discuss determining the cacheability of instructions during compilation and inserting a bit or other marking in the instructions corresponding to the determined cacheability.

The Morrison *et al.* patent also fails to suggest even that the basic concept of applicant's disclosed method and system. The Morrison *et al.* patent describes a technique for more efficiently executing branched instructions in a parallel computer system by re-ordering the instructions. The Morrison *et al.* patent teaches analyzing the instructions in a program during compilation based on the manner in which the instruction uses resources, such as whether to instructions are independent and can thus be executed in parallel. Each instruction is then assigned an execution time at step 130, *i.e.*, the amount of time required to execute the instruction. This execution time is referred to as the "instruction firing time" or "IFT." The instruction is also assigned other data at step 130 such as the identity of one of several processors that will execute the instruction. Finally, execution sets are built at step 140 by a re-ordering the instructions based upon the instruction firing time, the identity of the processor executing the instruction, etc.

The portion of the specification cited by the Examiner between column 27, line 56 and column 28, line 31 does not describe determining the cacheability of instructions during compilation or at any other time. Instead, this portion simply describes the components of the logical resource driver (LRD) shown in Figure 15 and the interconnections between those components. While the LRD shown in Figure 15 does include a data cache section 1500 and an instruction cache section 1512, the Morrison *et al.* patent does not provide any indication that the cacheability of either data or instructions are determined at compilation or any other time. Nor does the Morrison *et al.* patent indicate that a bit or other marking is added to an instruction to indicate the cacheability of the instruction so that the instruction can be cached accordingly during execution.

The remaining references cited in the Office Action fail to suggest the teachings of applicant's disclosed method and system that are missing from the teachings of the Serocki *et al.* patent and the Morrison *et al.* patent.

Turning, now, to the claims, claim 1, which has been rejected as being obvious over the Serocki *et al.* patent in view of the Morrison *et al.* patent, specifies a computer system having cache circuitry, a main memory and a processor controlled by a computer program. The processor directs selected portions of information related to a computer program stored in the cache circuitry to the cache circuitry "based at least in part on cacheability determinations made during compilation of the computer program." As explained above, neither the Serocki *et al.* patent nor the Morrison *et al.* patent describe making any cacheability determinations at either compilation or at any other time. Therefore, of necessity, they fail to disclose using cacheability determinations made during compilation for the purpose of directing selected portions of information to the cache circuitry, as recited in claim 1. Claim 1 is thus clearly patentable over the cited references.

Independent claim 25, which is directed to a system for determining which portions of a program code to cache and which not to cache, has been rejected as being anticipated by the Serocki *et al.* patent. The claimed system includes a processor connected to a memory device containing a program code. The processor is controlled by the program code "to direct selected portions of the program code to a cache based at least in part on cacheability determinations made during compilation of a computer program." As explained above, the

Serocki *et al.* patent is directed to branch prediction determinations; it does not describe or suggest making cacheability determinations during compilation or using the cacheability determinations to control caching during execution of the instructions.

Independent method claim 33 is directed to a method of controlling the cacheability of information in a computer system. This claim was also rejected as being anticipated by the Serocki *et al.* patent. However, it should be readily apparent that the Serocki *et al.* patent does not describe making cacheability determinations for information associated with a computer program that is being compiled, nor does it describe marking at least selected portions of information according to the cacheability determinations. Likewise, the Serocki *et al.* patent fails to teach detecting markings of selected portions of information during execution and directing the selected portions of information to the cache circuitry according to the markings. If the Examiner believes the subject matter of this claim is taught in the Serocki *et al.* patent, he is respectfully requested to specifically identify the column and line numbers where this subject matter is disclosed.

The final Independent claim is claim 48, which is directed to a method for compiling a computer program. The claimed method includes making cacheability determinations for information associated with a computer program and then marking at least selected portions of the information according to the determinations. Claim was also rejected as being anticipated by the Serocki *et al.* patent. However, as explained above, the Serocki *et al.* patent does not teach making cacheability determinations during compilation of a computer program and then marking information associated with the computer program according to the determinations.

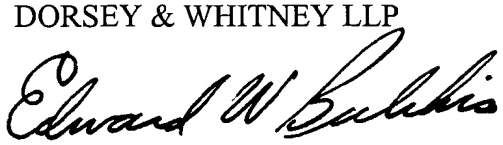
The remaining claims in the application patentably distinguish over the cited references because of their dependency on patentable independent claims and because of the additional limitations added by those claims.

The other matters raised by the Office Action are also being addressed in this response. Applicant is filing a new set of drawings in response to the drawing objection, the specification is being amended to correctly refer to step 318, and claim 45 is being amended to obviate the Section 112 rejection.

Insofar as this application and the drawings are now in proper form, and claims 1-61 patentably distinguish over the cited references, favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,

DORSEY & WHITNEY LLP



Edward W. Bulchis
Registration No. 26,847
Telephone No. (206) 903-8785

EWB:pep

Enclosures:
- Postcard
- Fee Transmittal Sheet (+ copy)
- Filing Formal Drawings
- 5 Sheets of Formal Drawings (Figs 1-5)

DORSEY & WHITNEY LLP
1420 Fifth Avenue, Suite 3400
Seattle, WA 98101-4010
(206) 903-8800 (telephone)
(206) 903-8820 (fax)

c:\ndeloc\temp.pogson.phoebe\500050.01 amend oa 042403.doc